C. R. Acad. Sci. Paris, Ser. I 348 (2010) 581–585

Contents lists available at ScienceDirect

C. R. Acad. Sci. Paris, Ser. I

www.sciencedirect.com

Numerical Analysis

A new class of fractional step techniques for the incompressible Navier–Stokes equations using direction splitting

Une nouvelle classe de techniques de pas fractionnaires basée sur les directions alternées pour les équations de Navier–Stokes incompressibles

Jean-Luc Guermond [a,1], Peter D. Minev [b]

[a] *Department of Mathematics, Texas A&M University, College Station, TX 77843-3368, USA*
[b] *Department of Mathematical and Statistical Sciences, University of Alberta, Edmonton, Alberta, Canada T6G 2G1*

ARTICLE INFO

Article history:
Received 14 February 2010
Accepted after revision 12 March 2010
Available online 1 April 2010

Presented by Roland Glowinski

ABSTRACT

A new direction-splitting-based fractional time stepping is introduced for solving the incompressible Navier–Stokes equations. The main originality of the method is that the pressure correction is computed by solving a sequence of one-dimensional elliptic problems in each spatial direction. The method is very simple to program in parallel, very fast, and has exactly the same stability and convergence properties as the Poisson-based pressure-correction technique, either in standard or rotational form.

RÉSUMÉ

Nous proposons une nouvelle famille d'algorithmes à pas fractionnaires basés sur la technique des directions alternées pour la résolution des équations de Navier–Stokes. Chaque étape de l'algorithme consiste à résoudre une série de problèmes uni-dimensionels. On démontre que la méthode est unconditionellement stable et convergente.

Version française abrégée

Il est maintenant bien connu que les méthodes classiques de pas fractionnaires pour l'approximation de (1) sont toutes des réalisations discrètes de la perturbation singulière (2), cf. e.g. [4–7]. Dès que (2) est en jeu, on est invariablement conduit au paradigme de projection de Chorin [1] et Temam [8] qui consiste à décomposer les champs de vecteurs en une composante solénoidale plus un gradient. Par exemple, en posant $u_\epsilon = \mathbf{v}_\epsilon + \epsilon\nabla\phi_\epsilon$, la seconde équation de (2) implique $\nabla\cdot\mathbf{v}_\epsilon = 0$ et $\mathbf{v}_\epsilon|_{\Gamma\times(0,T)} = 0$. Nous abandonnons ce paradigme en proposant de remplacer (2) par (3) où l'opérateur A est libre mais doit satisfaire les hypothèses (4). On se retrouve dans la situation classique dès que $-A$ est le Laplacien avec des conditions aux limites de Neumann homogènes. Les hypothèses (4) sont suffisantes pour démontrer que les propriétés de convergence de la paire (u_ϵ, p_ϵ), solution de (3), vers (u, p) sont les mêmes que pour la paires (u_ϵ, p_ϵ), solution de (2), pourvu que la paire (u, p) est suffisamment régulière. Lorsque le domaine Ω est un rectangle ou un parallépidède rectangle, deux réalisations intéressantes de A sont définies en (5), en deux dimensions d'espace, et (6) en trois dimensions d'espace,

E-mail addresses: guermond@math.tamu.edu (J.-L. Guermond), minev@ualberta.c (P.D. Minev).
[1] Permanent address: LIMSI (CNRS-UPR 3251), BP 133, 91403 Orsay, France.

1631-073X/$ – see front matter
doi:10.1016/j.crma.2010.03.009

voir le lemme 2.1. La propriété remarquable de l'opérateur ainsi défini est que la résolution du problème $Ap = f$ se réduit à la résolution d'une série de problèmes unidimensionnels très simples.

L'idée centrale de cette Note est d'utiliser (3) comme point de départ pour construire une nouvelle famille d'algorithmes à pas fractionnaires.

Lorsque le domaine Ω est quelconque nous proposons en (10)–(13) un algorithme à pas fractionnaires générique où l'opérateur A est arbitraire mais satisfait (4). Cet algorithme est une généralisation de l'algorithme de correction de pression classique basé sur la décomposition de Helmholtz. Le paramètre χ est à choisir dans l'intervalle $[0, 1]$. On dit que l'algorithme est sous forme standard lorsque $\chi = 0$ et sous forme rotationnelle sinon. La forme rotationnelle de l'algorithme est quasi-optimale en termes de taux de convergence alors que la forme standard est sous-optimale ; le lecteur est renvoyé à [4] pour les détails. Le théorème 3.1 montre que (10)–(13) a exactement les mêmes propriétés de stabilité que l'algorithme de correction de pression classique en forme standard. Un résultat similaire est vrai pour la forme rotationnelle, cf. [3].

Jusqu'ici l'idée d'introduire une nouvel opérateur A n'est pas très originale. Toutefois cette idée ouvre des perspectives radicalement nouvelles si le domaine Ω est un rectangle ou un parallépidède rectangle car on peut alors choisir l'opérateur A défini en (5) ou (6). En remplaçant la résolution monolithique de la conservation de la quantité de mouvement (11) par une méthode de directions alternées décrite en (15), cf. Douglas [2], l'algorithme (10)–(15), (12)–(13) devient alors très simple dès que l'opérateur A est défini par (5) ou (6). Les deux étapes (10) et (13) sont triviales et les deux étapes (15) et (12) consistent à résoudre une succession de problèmes elliptiques unidimensionnels. L'algorithme (10)–(15), (12)–(13) est stable, convergent, et d'ordre deux sur la vitesse en norme $\mathbf{L}^2$, cf. [3]. La version rotationnelle est stable pour $\chi = \frac{1}{2}$.

Nous avons testé numériquement les propriétés de convergence de l'algorithme (10)–(15), (12)–(13). Nous considérons $\Omega = (0, 1)^2$ et nous utilisons le schéma MAC pour l'approximation spatiale. La solution analytique est $\mathsf{u} =: (\sin x \sin(y + t), \cos x \cos(y + t))$, $\mathsf{p} := \cos x \sin(y + t)$ (le terme source $\mathbf{f}$ est défini en conséquence). La Fig. 1 montre la norme L^2 de l'erreur sur les deux composantes Cartésiennes de la vitesse en fonction du pas de temps Δt sur trois maillages différents (40×40 (ligne discontinue), 80×80 (ligne en pointillés) and 160×160 (ligne discontinue avec des pointillés)). Ces résultats confirment que l'algorithme est du second ordre en temps sur la vitesse dans la norme $\mathbf{L}^2$. Les mêmes tests de convergence sur la pression (non représentés ici) montrent aussi un taux de convergence voisin de deux, ce qui est meilleur que le taux $\mathcal{O}(\Delta t^{\frac{3}{2}})$ théorique, cf. [5,3].

Nous avons programmé une version parallèle de (10)–(15), (12)–(13) en trois dimensions d'espace en utilisant la méthode du complément de Schur pour résoudre les problèmes unidimensionnels. Nos tests montrent un taux de parallélisme quasi-optimal jusqu'à 1024 processeurs (1024 est le nombre maximal de processeurs qu'il est possible d'utiliser sans autorisation spéciale sur le cluster Hurr de IAMCS à Texas A&M). A titre d'exemple nous montrons dans la Table 1 le temps CPU par pas de temps pour differents maillages et differents nombres de processeurs : 1, 512, et 1024. On considère trois maillages : 2.7×10^4 noeuds par processeur, 2.16×10^5 noeuds par processeur, 10^6 noeuds par processeur. Nous observons qu'à nombre de noeuds par processeur fixé, le temps CPU ne croît pas de façon significative avec le nombre de processeur. Nous conjecturons que la méthode reste quasi-optimale avec plusieurs dizaines de milliers de processeurs.

1. Introduction

This Note deals with the numerical approximation of the time-dependent Navier–Stokes equations on a finite time interval $[0, T]$ and in a square or cubic domain, $\Omega = (0, 1)^d$, $d = 2, 3$. Since the nonlinearity in the Navier–Stokes equations and the incompressibility constraint do not interfere, we henceforth restrict the discussion to the time-dependent Stokes equations:

$$\begin{cases} \partial_t \mathsf{u} - \nu \Delta \mathsf{u} + \nabla \mathsf{p} = \mathbf{f} \quad \text{in } \Omega \times (0, T), \\ \nabla \cdot \mathsf{u} = 0 \quad \text{in } \Omega \times (0, T), \\ \mathsf{u}|_\Gamma = 0 \quad \text{in } (0, T), \quad \text{and} \quad \mathsf{u}|_{t=0} = \mathsf{u}_0 \quad \text{in } \Omega, \end{cases} \tag{1}$$

where $\mathbf{f}$ is a smooth source term and u_0 is a solenoidal initial velocity field with zero normal trace.

Approximating the coupled system (1) is computer intensive due to the elliptic character induced by the incompressibility constraint. Once time is discretized and if the velocity/pressure coupling is retained, (1) reduces to solving a generalized Stokes problem at each time step. Alternative strategies for solving (1) consists of uncoupling the velocity and the pressure by using the so-called projection algorithms. These techniques have been used in computational fluid dynamics for about four decades starting with the pioneering works of Chorin [1] and Temam [8]. They underwent some evolution in the 1980s and 1990s, but the same fundamental idea of decomposing vector fields into a divergence-free part and a gradient has remained unchanged. The total cost per time step is that of solving one vector-valued advection-diffusion equation and one scalar-valued Poisson equation with homogeneous Neumann boundary condition. For large size problems and large Reynolds numbers, the cost of solving the Poisson equation becomes dominant.

The goal of the present Note is to address the Poisson equation issue by proposing a new method to reduce the computational complexity of the enforcement of the incompressibility constraint. The projection paradigm in which vector fields are decomposed into a divergence-free part and a gradient part is abandoned. The new key idea consists of replacing the standard Poisson problem for the pressure correction by a succession of one-dimensional second-order boundary value problems in each spatial direction. This technique is proved to be unconditionally stable and convergent. Numerical tests show that the incremental version of the method is second-order accurate in time on the velocity in the $\mathbf{L}^2$-norm.

2. The heuristics

It is now well understood that most of the incremental pressure-correction algorithms, see e.g. [4–7], are discrete realizations of the following $\mathcal{O}(\epsilon^2)$ perturbation of (1):

$$\begin{cases} \partial_t \mathsf{u}_\epsilon - \nu\Delta \mathsf{u}_\epsilon + \nabla \mathsf{p}_\epsilon = \mathbf{f} \quad \text{in } \Omega\times[0,T], & \mathsf{u}_\epsilon|_{\Gamma\times[0,T]} = 0, \quad \mathsf{u}_\epsilon|_{t=0} = \mathbf{u}_0 \\ -\epsilon\Delta\phi_\epsilon + \nabla\cdot\mathsf{u}_\epsilon = 0 \quad \text{in } \Omega\times[0,T], & \partial_n\phi_\epsilon|_{\Gamma\times[0,T]} = 0, \\ \epsilon\partial_t \mathsf{p}_\epsilon = \phi_\epsilon - \chi\nu\nabla\cdot\mathsf{u}_\epsilon & \mathsf{p}_\epsilon|_{t=0} = \mathsf{p}_0, \end{cases} \tag{2}$$

where the parameter $\chi\in[0,1]$ is user-dependent and $\mathsf{p}_0 := \mathsf{p}(t{=}0)$. We are now going to use (2) as the starting point for a new solution method. Actually, the analysis of (2) reveals that, modulo appropriate regularity assumptions, the convergence properties of $\mathsf{u}_\epsilon, \mathsf{p}_\epsilon$ are unchanged by generalizing (2) as follows:

$$\begin{cases} \partial_t \mathsf{u}_\epsilon - \nu\Delta \mathsf{u}_\epsilon + \nabla \mathsf{p}_\epsilon = \mathbf{f} \quad \text{in } \Omega\times[0,T], & \mathsf{u}_\epsilon|_{\Gamma\times[0,T]} = 0, \quad \mathsf{u}_\epsilon|_{t=0} = \mathbf{u}_0, \\ \epsilon A\phi_\epsilon + \nabla\cdot\mathsf{u}_\epsilon = 0 \quad \text{in } \Omega\times[0,T], & \phi_\epsilon(t)\in D(A), \\ \epsilon\partial_t \mathsf{p}_\epsilon = \phi_\epsilon - \chi\nu\nabla\cdot\mathsf{u}_\epsilon & \mathsf{p}_\epsilon|_{t=0} = \mathsf{p}_0. \end{cases} \tag{3}$$

The key generic properties that are required on the operator A for $\mathbf{u}_\epsilon$ and p_ϵ to converge to u and p as $\epsilon\to 0$ are that the domain $D(A)$ and the bilinear form $a(p,q) := \int_\Omega qAp\,\mathrm{d}\mathbf{x}$ satisfy the following assumptions:

$$a \text{ is symmetric}, \quad \text{and} \quad \|\nabla q\|_{\mathbf{L}^2}^2 \leqslant a(q,q), \quad \forall q\in D(A). \tag{4}$$

Let

$$\begin{cases} A := (1-\partial_{xx})(1-\partial_{yy}) \\ D(A) := \big\{p,\ (1-\partial_{yy})p,\ Ap\in L^2(\Omega),\ p|_{y=0,1} = 0,\ \partial_x\big((1-\partial_{yy})p\big)\big|_{x=0,1} = 0\big\} \end{cases} \tag{5}$$

in two space dimensions, and

$$\begin{cases} A := (1-\partial_{xx})(1-\partial_{yy})(1-\partial_{zz}) \\ D(A) := \big\{p,\ (1-\partial_{zz})p,\ (1-\partial_{yy})(1-\partial_{zz})p,\ Ap\in L^2(\Omega), \\ \qquad p|_{z=0,1} = 0,\ \partial_y\big((1-\partial_{zz})p\big)\big|_{y=0,1} = 0,\ \partial_x\big((1-\partial_{yy})(1-\partial_{zz})p\big)\big|_{x=0,1} = 0\big\} \end{cases} \tag{6}$$

in three space dimensions. One interesting property of this operator is that solving $Ap = f$ for all $f\in L^2(\Omega)$ is elementary. For instance solving $Ap = f$ in three space dimension amounts to solving a succession of three one-dimensional problems as follows: find ψ, φ, and p so that

$$\psi - \partial_{xx}\psi = f, \quad \partial_x\psi|_{x=0,1} = 0; \tag{7}$$

$$\varphi - \partial_{yy}\varphi = \psi, \quad \partial_y\varphi|_{y=0,1} = 0; \tag{8}$$

$$p - \partial_{zz}p = \varphi, \quad \partial_z p|_{z=0,1} = 0. \tag{9}$$

Another important property of A is summarized in the following:

Lemma 2.1. *Consider $A : D(A)\subset L^2(\Omega)/\mathbb{R}\to L^2(\Omega)/\mathbb{R}$ be defined by* (5) *or* (6)*, in two or three space dimensions, respectively. The bilinear form $a(p,q) := \int_\Omega qAp\,\mathrm{d}\mathbf{x}$ satisfies* (4)*.*

The novel idea described in this Note consists of using (3) and Lemma 2.1 to construct a new family of very efficient pressure-correction algorithms.

3. A generalized pressure-correction algorithm

Let $A : D(A)\subset L^2(\Omega)/\mathbb{R}\to L^2(\Omega)/\mathbb{R}$ be an unbounded closed operator satisfying the assumptions (4). We now describe a generalized pressure-correction algorithm based on (3).

Pressure predictor: The algorithm is initialized by setting $p^{-\frac{1}{2}} = p^{-\frac{3}{2}} = \mathsf{p}_0$, and a pressure predictor is computed as follows for $n\geqslant 0$:

$$p^{*,n+\frac{1}{2}} = 2p^{n-\frac{1}{2}} - p^{n-\frac{3}{2}}. \tag{10}$$

Velocity update: The velocity is initialized by setting $\mathbf{u}^0 = \mathsf{u}_0$. Using the Crank–Nicolson technique to approximate the momentum equation and $p^{*,n+\frac{1}{2}}$ as a pressure predictor, the velocity $\mathbf{u}^{n+1}$, $n\geqslant 0$, is updated by solving

$$\frac{1}{\Delta t}\big(\mathbf{u}^{n+1}-\mathbf{u}^{n}\big)-\frac{1}{2}\nu\Delta\big(\mathbf{u}^{n+1}+\mathbf{u}^{n}\big)+\nabla p^{*,n+\frac{1}{2}}=\mathbf{f}\big(t^{n+\frac{1}{2}}\big),\quad \mathbf{u}^{n+1}|_{\Gamma}=0. \tag{11}$$

Pressure-correction calculation: A pressure correction $\phi^{n+\frac{1}{2}}\in D(A)$ is computed by solving:

$$A\phi^{n+\frac{1}{2}}=-\frac{1}{\Delta t}\nabla\cdot\mathbf{u}^{n+1}. \tag{12}$$

Pressure update: Finally the pressure is updated as follows:

$$p^{n+\frac{1}{2}}=p^{n-\frac{1}{2}}+\phi^{n+\frac{1}{2}}-\chi\nu\nabla\cdot\left(\frac{1}{2}\big(\mathbf{u}^{n+1}+\mathbf{u}^{n}\big)\right). \tag{13}$$

Let us define the norm $\|q\|_A=a(q,q)^{\frac{1}{2}}$, for all q in $D(A)$. The main result of this Note is the following:

Theorem 3.1. *The solution to* (10)–(13)*, with* $p^{-\frac{1}{2}}=p^{-\frac{3}{2}}=p_0$ *and* $\chi=0$*, satisfies the following stability estimate for all* $n\geqslant 0$*:*

$$\big\|\mathbf{u}^{n+1}\big\|_{\mathbf{L}^2}^2+\Delta t^2\big\|p^{n+\frac{1}{2}}\big\|_A^2+\nu\Delta t\sum_{k=0}^{n}\left\|\frac{1}{2}\nabla\big(\mathbf{u}^{k+1}+\mathbf{u}^{k}\big)\right\|_{\mathbf{L}^2}^2\leqslant\big\|\mathbf{u}^{0}\big\|_{\mathbf{L}^2}^2+\Delta t^2\big\|p^{-\frac{1}{2}}\big\|_A^2+\frac{\Delta t\nu}{2}\big\|\nabla\mathbf{u}_0\big\|_{\mathbf{L}^2}^2. \tag{14}$$

The major novelty of the algorithm (10)–(13) is that the Poisson equation of the usual pressure-correction method (which was inherited from the projection paradigm) has been replaced by an abstract operator A which can now take many forms. When Ω is a square or a cube, a very interesting realization of this operator is (5) in two space dimensions and (6) in three space dimensions. This choice replaces the computationally intensive solution of a Poisson problem by a succession of trivial one-dimensional elliptic problems. The computational savings can be very significant, in particular when the velocity update is also done by a direction splitting technique.

The Crank–Nicolson time stepping is nonessential in (11). Other time marching schemes can be used to approximate the momentum equation, say for instance second-order backward finite differences.

4. A fully split pressure-correction algorithm

We now propose a fully split pressure-correction algorithm for which the velocity update is done by using a direction splitting technique proposed by Douglas [2]. The algorithm is initialized by setting $p^{-\frac{1}{2}}=p^{-\frac{3}{2}}=\mathrm{p}_0$, and the pressure predictor $p^{*,n+\frac{1}{2}}$ is defined by (10). The velocity is initialized by setting $\mathbf{u}^0=\mathrm{u}_0$, and for $n\geqslant 0$, $\mathbf{u}^{n+1}$ is updated by solving

$$\begin{aligned}
&\frac{\boldsymbol{\xi}^{n+1}-\mathbf{u}^{n}}{\Delta t}-\nu\Delta\mathbf{u}^{n}+\nabla p^{*,n+\frac{1}{2}}=\mathbf{f}\big(t^{n+\frac{1}{2}}\big),\quad \mathbf{u}^{n}\big|_{\Gamma}=0,\\
&\frac{\boldsymbol{\eta}^{n+1}-\boldsymbol{\xi}^{n+1}}{\Delta t}-\frac{\nu}{2}\partial_{xx}\big(\boldsymbol{\eta}^{n+1}-\mathbf{u}^{n}\big)=0,\qquad \boldsymbol{\eta}^{n+1}\big|_{x=0,1}=0,\\
&\frac{\boldsymbol{\zeta}^{n+1}-\boldsymbol{\eta}^{n+1}}{\Delta t}-\frac{\nu}{2}\partial_{yy}\big(\boldsymbol{\zeta}^{n+1}-\mathbf{u}^{n}\big)=0,\qquad \boldsymbol{\zeta}^{n+1}\big|_{y=0,1}=0,\\
&\frac{\mathbf{u}^{n+1}-\boldsymbol{\zeta}^{n+1}}{\Delta t}-\frac{\nu}{2}\partial_{zz}\big(\mathbf{u}^{n+1}-\mathbf{u}^{n}\big)=0,\qquad \mathbf{u}^{n+1}\big|_{z=0,1}=0.
\end{aligned} \tag{15}$$

The two-dimensional version of the algorithm is obtained by omitting the last step and setting $\mathbf{u}^{n+1}=\boldsymbol{\zeta}^{n+1}$. The pressure correction $\phi^{n+\frac{1}{2}}$ is computed by solving (12) with A defined in (5) or (6), depending on the space dimension. The pressure is updated using (13). Numerical tests reveal that the rotational form of the algorithm is stable only for $\chi\leqslant\frac{1}{2}$ and works best for $\chi=\frac{1}{2}$.

5. Numerical results

We report in this section convergence tests illustrating the performance of the fully split algorithm (10)–(15), (12)–(13) with $\chi=\frac{1}{2}$. The tests are made in two space dimensions using second-order central differences on a uniform MAC stencil and the following solution to the unsteady Stokes equations $\mathrm{u}=:(\sin x\sin(y+t),\cos x\cos(y+t))$, $\mathrm{p}:=\cos x\sin(y+t)$ (the source term $\mathbf{f}$ is set appropriately). Fig. 1 shows the L^2-norm of the error on the two Cartesian components of the velocity as a function of Δt on three different meshes (40×40 (dashed line), 80×80 (dotted line) and 160×160 (dash-dotted line)). These results clearly confirm that the fully split pressure-correction algorithm is second-order accurate on the velocity in the $\mathbf{L}^2$-norm. The results on the pressure (not shown here) demonstrate nearly second-order accuracy in time which is better than the expected $\frac{3}{2}$ rate, cf. [5].

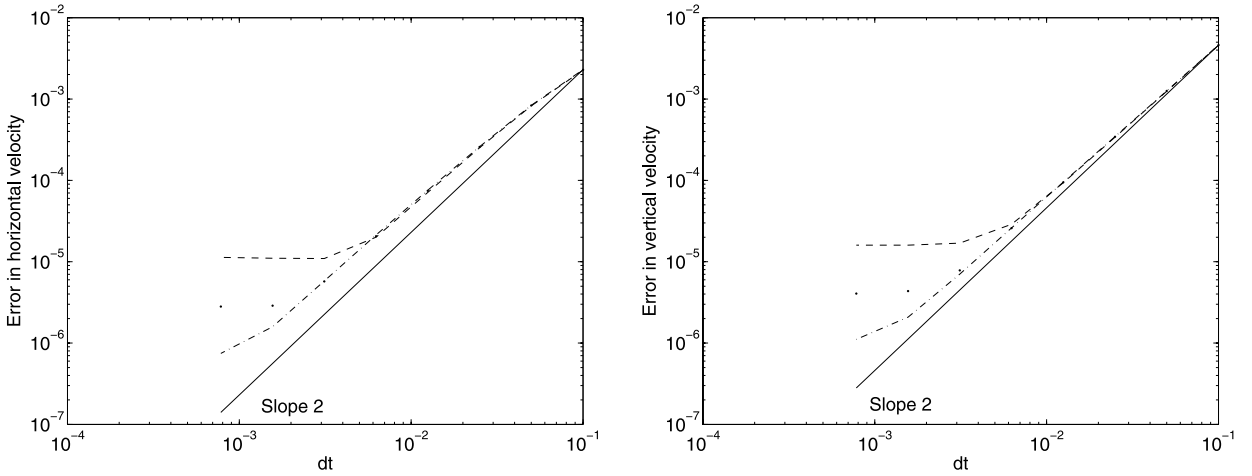

Fig. 1. L^2-norm of the error on the horizontal (left column) and vertical (right column) components of the velocity with the rotational incremental algorithm at $t = 2$ on three grids, 40×40 (dashed line), 80×80 (dotted line) and 160×160 (dash-dotted line).

Table 1
Parallel performance: CPU time (in second) per time step for (10)–(15), (12)–(13) + explicit nonlinear terms.

# procs	2.7×10^4 nds/proc	2.16×10^5 nds/proc	10^6 nds/proc
1	0.17 s	1.19 s	5.54 s
512	0.20 s	1.29 s	5.86 s
1024	0.23 s	1.36 s	5.93 s

The authors have programmed a parallel version of (the Navier–Stokes version of) (10)–(15), (12)–(13) in three space dimensions using a direct method based on a Schur complement technique to solve the one-dimensional problems. Preliminary tests show perfect parallel scaling up to the maximum number of processors that were available to the authors without special request on the Hurr HPC cluster of IAMCS at Texas A&M (1024 processors), see Table 1. For instance, the 3D driven cavity problem of aspect ratio 1:1:2 at $R_e = 1000$ solved up to time $T = 12$ on 1024 processors with 10^6 nodes per processor takes 5.3 s per time step. Larger computations are currently being performed to provide a benchmark solution to this problem.

The theory developed in this Note holds only if Ω is a cube. This limitation is serious, but many fundamental problems can be formulated in a cube: differentially heated cavities, driven cavities, turbulent pipe and channel flows, Rayleigh–Taylor and Rayleigh–Benard instabilities, etc. We conjecture that any problem that can be smoothly mapped into a cube is solvable with the direction splitting described above. More complicated geometries can be treated by using the fictitious domain technique.

Acknowledgements

This material is based upon work supported by the National Science Foundation grants DMS-0713829. This publication is also partially based on work supported by Award No. KUS-C1-016-04, made by King Abdullah University of Science and Technology (KAUST). The work of P. Minev is also supported by fellowships from the Institute of Applied Mathematics and Computational Science and the Institute of Scientific Computing at Texas A&M University and a Discovery grant of NSERC.

References

[1] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comp. 22 (1968) 745–762.
[2] J. Douglas Jr., Alternating direction methods for three space variables, Numer. Math. 4 (1962) 41–63.
[3] J.-L. Guermond, P. Minev, A. Salgado, Convergence analysis of new class of direction splitting algorithm for the Navier–Stokes equations, 2010, in preparation.
[4] J.-L. Guermond, P. Minev, J. Shen, An overview of projection methods for incompressible flows, Comput. Methods Appl. Mech. Engrg. 195 (2006) 6011–6054.
[5] J.L. Guermond, Jie Shen, On the error estimates for the rotational pressure-correction projection methods, Math. Comp. 73 (248) (2004) 1719–1737 (electronic).
[6] R. Rannacher, On Chorin's Projection Method for the Incompressible Navier–Stokes Equations, Lecture Notes in Mathematics, vol. 1530, 1992.
[7] J. Shen, On error estimates of projection methods for the Navier–Stokes equations: second-order schemes, Math. Comp. 65 (215) (July 1996) 1039–1065.
[8] R. Temam, Sur l'approximation de la solution des équations de Navier–Stokes par la méthode des pas fractionnaires ii, Arch. Rat. Mech. Anal. 33 (1969) 377–385.